

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук _____

(повна назва)

Кафедра _____ програмної інженерії _____

(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Програмна система для управління заходами

(тема)

Виконав:

здобувач _____ року навчання
групи ПЗП-21-4

Максим ГАЛУШКА

(Власне ім'я, ПРІЗВИЩЕ)

Спеціальність 121 – Інженерія програмного
забезпечення

(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна _____

Освітня програма Програмна інженерія

(повна назва освітньої програми)

Керівник доцент кафедри ПІ Ольга ВОРОЧЕК

(посада, Власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту

Зав. кафедри

Кирило СМЕЛЯКОВ

(підпис) (Власне ім'я, ПРІЗВИЩЕ)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук
 Кафедра _____ програмної інженерії
 Рівень вищої освіти _____ перший (бакалаврський)
 Спеціальність _____ 121 – Інженерія програмного забезпечення
 Тип програми _____ Освітньо-професійна
 Освітня програма _____ Програмна Інженерія
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«___» _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувач _____ Галушка Максим Віталійович
 (прізвище, ім'я, по батькові)

1. Тема роботи _____ Програмна система для управління заходами

Затверджена наказом по університету від 19 травня 2025р. №397Ст

2. Термін подання здобувачом роботи до екзаменаційної комісії 10.06.2024

3. Вихідні дані до роботи Розробити програмну систему з управління заходами, серверна частина якої реалізована за допомогою фреймворку NestJS з використанням мови TypeScript та бази даних PostgreSQL. Клієнтська частина створена на базі Next.js, забезпечуючи взаємодію з REST API для виконання основних функцій.

4. Перелік питань, що потрібно опрацювати в роботі

Вступ, аналіз предметної галузі, формування вимог до програмної системи, архітектура та проектування програмного забезпечення, опис прийнятих програмних рішень, тестування розробленого програмного забезпечення, висновки, додатки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	29.03.2025	<i>виконано</i>
2	Створення специфікації ПЗ	17.04.2025	<i>виконано</i>
3	Проектування ПЗ	19.05.2025	<i>виконано</i>
4	Розробка ПЗ	25.05.2025	<i>виконано</i>
5	Тестування ПЗ	28.05.2025	<i>виконано</i>
6	Оформлення пояснювальної записки	05.06.2025	<i>виконано</i>
7	Підготовка презентації та доповіді	06.06.2025	<i>виконано</i>
8	Попередній захист	06.06.2025	<i>виконано</i>
9	Нормоконтроль, рецензування	07.06.2025	<i>виконано</i>
10	Здача роботи у електронний архів	07.06.2025	<i>виконано</i>
11	Допуск до захисту у зав. кафедри	10.06.2025	<i>виконано</i>

Дата видачі завдання 27 січня 2025р.

Здобувач _____

(підпис)

Галушка М.В.

Керівник роботи _____

(підпис)

доцент кафедри ІІ Ольга ВОРОЧЕК

(посада, Власне ім'я, ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи бакалавра: 54 с., 23 рис., 2 табл., 18 джерел.

ЗАХІД, КВИТОК, УПРАВЛІННЯ, КОРИСТУВАЧ, СТАТИСТИКА.

Об'єктом дослідження є процес організації та управління заходами, включаючи створення подій, реєстрацію учасників, продаж квитків, надсилання сповіщень і моніторинг активності.

Метою кваліфікаційної роботи є розробка веб-системи для управління заходами, яка дозволяє організаторам ефективно контролювати всі етапи підготовки та проведення подій, зокрема створення подій, управління користувачами, аналітику продажів квитків, перегляд статистики та збір відгуків.

Методи розробки базуються на сучасному стеку технологій, що включає Nest.js та мову програмування TypeScript для розробки серверної частини системи, а також фреймворк Next.js з TypeScript для створення клієнтського інтерфейсу. Для зберігання інформації використовується реляційна база даних PostgreSQL [11]. В системі реалізовано модулі для реєстрації користувачів, створення та перегляду заходів, оформлення квитків, залишення відгуків, а також панель адміністратора для контролю за подіями та статистикою. Впроваджено механізми автентифікації та авторизації, що забезпечують безпеку взаємодії користувачів з системою. Розгортання front-end і back-end частин виконано на платформі AWS, що гарантує високу доступність і масштабованість. У результаті роботи створено повноцінну програмну систему для автоматизованого управління заходами з можливістю масштабування та розширення функціоналу.

ABSTRACT

EVENT, TICKET, MANAGEMENT, USER, STATISTICS

The object of the research is the process of organizing and managing events, including event creation, participant registration, ticket sales, notification sending, and activity monitoring.

The goal of the thesis is to develop a web-based system for event management that enables organizers to efficiently control all stages of event preparation and execution, including event creation, user management, ticket sales analytics, statistics viewing, and feedback collection.

The development methods are based on a modern technology stack, including Nest.js and the TypeScript programming language for the server-side development, as well as the Next.js framework with TypeScript for building the client interface. A relational PostgreSQL database is used for data storage. The system includes modules for user registration, event creation and viewing, ticket issuance, feedback submission, and an admin panel for monitoring events and statistics. Authentication and authorization mechanisms have been implemented to ensure secure user interaction with the system. The deployment of the front-end and back-end parts has been carried out on the AWS platform, ensuring high availability and scalability.

As a result, a fully functional software system has been developed for automated event management, with scalability and extensibility capabilities.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної галузі.....	8
1.1 Аналіз предметної галузі.....	8
1.1.1 Огляд сучасних методів управління заходами.....	9
1.1.2 Бізнес-цілі та критерії успіху.....	10
1.1.3 Аналіз існуючих рішень.....	11
1.2 Виявлення та вирішення проблем.....	12
1.3 Постановка задачі.....	13
1.3.1 Цільова аудиторія.....	14
2 Формування вимог до програмної системи.....	15
3 Архітектура та проєктування програмного забезпечення.....	17
3.1 UML проєктування ПЗ.....	17
3.2 Проєктування структури зберігання даних.....	18
3.3 Приклади найцікавіших алгоритмів та методів.....	21
3.4 Створення UI/UX.....	22
4 Опис прийнятих програмних рішень.....	29
5 Тестування програмного забезпечення.....	39
Висновки.....	43
Перелік джерел посилання.....	44
Додаток А Результат перевірки роботи на плагіат.....	47
Додаток Б Слайди презентації.....	48

ВСТУП

У сучасному світі, де події та заходи стають важливою частиною соціального, культурного та ділового життя, ефективне управління ними набуває особливого значення. Організація подій вимагає координації багатьох процесів від планування та розсилки запрошень до реєстрації учасників, продажу квитків, збору аналітичних даних та комунікації з аудиторією. Зростання масштабів заходів, а також зростаючі очікування з боку учасників потребують впровадження сучасних цифрових рішень, здатних автоматизувати та оптимізувати ці процеси.

Метою роботи є розробка програмної системи для управління заходами веб платформи, що надає організаторам інструменти для повноцінного управління подіями. Система дозволяє створювати заходи, керувати інформацією про них, реєструвати учасників, продавати квитки, здійснювати відстеження активності, надсилати повідомлення, а також переглядати статистику.

Розроблена система має забезпечити простоту у використанні, гнучкість, розширюваність та безпеку, що є важливими критеріями для сучасних інформаційних систем. Актуальність теми зумовлена необхідністю цифрової трансформації у сфері організації подій, особливо з огляду на зростання популярності онлайн- і гібридних форматів заходів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі

Організації заходів стрімко розвивається, що спричиняє потребу в інструментах, здатних забезпечити якісне управління процесами, пов'язаними з плануванням та проведенням подій. У практиці організаторів все частіше використовуються цифрові рішення, оскільки традиційні методи, засновані на ручному введенні даних, втрачають ефективність. Зростання масштабів подій, збільшення кількості учасників та потреба в оперативній комунікації ставлять нові вимоги до інформаційних систем, що використовуються у цій галузі.

На ринку вже існує низка сервісів, що частково або повністю автоматизують процеси створення подій, реєстрації учасників, продажу квитків, аналітики тощо. Проте більшість із них мають обмеження, зокрема високу вартість, відсутність локалізації, складність у використанні або закриту архітектуру, що унеможлиблює адаптацію під специфічні потреби організаторів.

Зростає актуальність використання веб-застосунків, які не потребують встановлення додаткового програмного забезпечення та можуть бути доступними з будь-якого пристрою [1]. У зв'язку з цим зростає інтерес до розробки веб-систем, які дозволяють організаторам ефективно керувати усіма аспектами заходів — від створення і редагування подій до моніторингу статистики та зворотного зв'язку з користувачами. Особливо важливим є забезпечення зручного інтерфейсу, безпеки даних, підтримки ролей користувачів, автоматичної обробки платежів, а також інтеграції з зовнішніми сервісами. Створення подібної системи дозволить не лише спростити роботу організаторів, але й підвищити якість обслуговування учасників, забезпечуючи їм зручний і швидкий доступ до всіх необхідних функцій.

1.1.1 Огляд сучасних методів управління заходами

Управління заходами сьогодні значно змінюється завдяки технологіям. Традиційні способи організації подій, коли організатори використовували паперові реєстраційні форми і ручне введення даних, поступово витісняються інтегрованими цифровими платформами, які забезпечують автоматизацію численних процесів. Веб-системи і мобільні додатки стали важливими інструментами для організаторів заходів, надаючи їм можливість централізовано управляти всім циклом події — від реєстрації учасників до збору платежів і аналізу результатів [2].

Одним з основних сучасних методів є використання онлайн-платформ для створення, управління та моніторингу подій. Ці системи дозволяють організаторам створювати події, планувати розклад, продавати квитки, вести реєстрацію учасників та зберігати всю інформацію в одному місці. Це значно спрощує процес і знижує ймовірність помилок, які часто виникають при ручному введенні даних або використанні різних програмних засобів для кожного етапу.

Також великим попитом користуються платформи для збору зворотного зв'язку та аналітики. За допомогою таких інструментів організатори можуть отримати відгуки від учасників заходу та оцінити його успішність. Це дозволяє своєчасно коригувати процеси і підвищувати якість проведення подій у майбутньому. Оцінка відгуків та статистики допомагає аналізувати. Управління командою організаторів також стало значно зручнішим завдяки використанню платформ для управління проектами. Вони дозволяють ефективно розподіляти завдання серед членів команди, вести облік виконання та забезпечувати прозору комунікацію між усіма учасниками процесу.

Завдяки таким методам і інструментам управління, організатори можуть значно підвищити ефективність своєї роботи, знижуючи витрати часу і ресурсів, а також забезпечуючи високу якість обслуговування учасників.

Сучасні технології допомагають оптимізувати процеси, зменшити людський фактор, мінімізувати помилки та створювати більш гнучке управління.

1.1.2 Бізнес-цілі та критерії успіху

Основною бізнес-ціллю розробки програмної системи для управління заходами є створення ефективного інструменту для автоматизації процесів організації подій. Система повинна допомогти організаторам знизити витрати часу та ресурсів на адміністрування, реєстрацію учасників, продаж квитків та взаємодію з відвідувачами. Важливо також забезпечити можливість моніторингу подій у реальному часі та збір аналітики для покращення процесів.

Основні бізнес-цілі включають автоматизацію управлінських процесів, створення єдиного інтерфейсу для управління подіями, що дозволить організаторам швидко налаштовувати заходи, реєструвати учасників та обробляти платежі без використання сторонніх сервісів. Система має гарантувати зручність для користувача, даючи можливість учасникам швидко реєструватися, купувати квитки та отримувати актуальну інформацію. Крім того, ключовим завданням є оптимізація маркетингу завдяки підключенню соцмереж і платіжних сервісів, що сприятиме зростанню аудиторії.

Крім того, система має бути масштабованою, щоб задовольняти потреби різних типів заходів, від невеликих зустрічей до великих міжнародних конференцій. Критерії успіху цієї системи включають підвищення задоволення користувачів, зниження витрат часу, збільшення кількості учасників завдяки ефективним маркетинговим інструментам та забезпечення аналітики та зворотного зв'язку для покращення якості заходів. Система також повинна легко інтегруватися з іншими платформами та сервісами для розширення її функціональних можливостей.

1.1.3 Аналіз існуючих рішень

Існує багато різноманітних рішень для управління заходами, які пропонують організаторам різні функціональні можливості для автоматизації процесів. Проте, не всі з них відповідають вимогам щодо ефективності, зручності користування або можуть забезпечити повний спектр послуг для організації подій різного масштабу.

Одним із найбільш популярних рішень є платформи для управління подіями, такі як Eventbrite, Ticketmaster, Bizzabo. Вони дозволяють організаторам створювати події, продавати квитки, управляти учасниками та збором коштів. Наприклад, Eventbrite надає можливість організаторам створювати події, продавати квитки, здійснювати маркетингові кампанії через електронні листи, а також збирати відгуки від учасників. Проте, обмеження на налаштування та інтеграції з іншими системами може бути перешкодою для організацій з більш складними вимогами. Ticketmaster спеціалізується на продажу квитків для великих заходів, зокрема концертів і спортивних подій, надаючи велику кількість платіжних опцій та інструментів для моніторингу продажів. Однак ця платформа не забезпечує таких можливостей для управління подіями як інші платформи, зокрема для невеликих заходів або внутрішніх корпоративних подій. Bizzabo, в свою чергу, орієнтована на великий бізнес і є однією з найбільш комплексних платформ, яка надає інструменти для організації не тільки продажу квитків, але й для управління подіями, мережевими заходами та аналітики. Однак її вартість та складність можуть бути бар'єром для малого та середнього бізнесу.

Попри існуючі варіанти, жодне з цих рішень не забезпечує повної гнучкості та можливості для організаторів створювати платформу, яка б відповідала усім їхнім індивідуальним вимогам, таким як специфічні функціональні можливості або інтеграція з іншими програмними продуктами. Крім того, більшість з цих рішень має обмеження в плані адаптації до специфічних потреб різних типів заходів. Таким чином, попит на більш гнучкі

та інтегровані рішення для управління заходами, що враховують потреби організаторів та учасників, залишається високим.

1.2 Виявлення та вирішення проблем

Організатори подій, незалежно від розміру заходу, часто стикаються з численними труднощами, що виникають під час планування, організації та моніторингу подій. Однією з найбільших проблем є відсутність централізованої системи для управління всіма аспектами заходу, що змушує використовувати кілька різних інструментів для виконання різних завдань, таких як реєстрація учасників, продаж квитків, збір даних, а також сповіщення учасників. Внаслідок цього виникають значні простой, зростає навантаження на обробку інформації, а також неминучі похибки, обумовлені людським фактором чи розбіжностями між системами.

Багато процесів ще виконуються вручну або за допомогою неповних інструментів, що ускладнює контроль за виконанням завдань і додає значні труднощі у взаємодії між учасниками, організаторами та іншими зацікавленими сторонами. Наприклад, використання окремих платформ для обробки реєстрацій, платіжних операцій і сповіщень створює хаос в управлінні даними. Це веде до втрати часу, неузгодженості даних, дублювання роботи та труднощів у здійсненні точного обліку. Інформація, яка накопичується в різних місцях, часто є неповною або неактуальною, що може ускладнити прийняття рішень.

Ще однією проблемою є відсутність зручної та ефективної системи для аналізу даних, що поступають, і для збору зворотного зв'язку від учасників. Без надійних інструментів моніторингу організатори не можуть оперативно оцінити ефективність заходу, зрозуміти, які аспекти потребують покращення, чи належним чином реагувати на потреби учасників під час проведення події. Це може призвести до зниження якості обслуговування і навіть негативно

вплинути на репутацію організаторів.

Для ефективного вирішення цих проблем необхідно розробити єдину веб-систему для управління заходами. Така система повинна об'єднувати в собі всі основні функції, пов'язані з організацією подій: реєстрацію учасників, продаж квитків, обробку платежів, сповіщення учасників про зміни або нагадування, а також можливість моніторингу та аналізу подій у реальному часі [9]. Автоматизація цих процесів не лише дозволить знизити кількість помилок і зайвих витрат часу, а й зробить взаємодію між організаторами і учасниками більш зручною та ефективною. Впровадження такої системи забезпечить високу точність даних, оперативне реагування на запити користувачів та можливість швидкої адаптації до змін під час заходу.

1.3 Постановка задачі

Завданням проекту є розробка онлайн-системи керування подіями, яка спростить основні етапи організації заходів і поліпшить взаємодію між усіма учасниками процесу. Впровадження системи включатиме такі етапи роботи:

- створити зручну систему реєстрації та авторизації користувачів;
- розробити функціонал для створення, редагування та видалення подій;
- реалізувати механізм реєстрації учасників і продажу квитків;
- впровадити засоби сповіщення користувачів про зміни або нагадування;
- додати можливості для перегляду статистики та аналітики подій;
- окрему панель адміністратора для управління всією системою;
- забезпечити зручний і безпечний інтерфейс користувача.

Виконання зазначених задач дозволить створити ефективний інструмент, що спростить процес організації заходів і забезпечить високий рівень обслуговування учасників.

1.3.1 Цільова аудиторія

Цільовою аудиторією програмної системи є організатори різноманітних заходів, таких як конференції, тренінги, виставки, концерти та спортивні події, а також приватні особи та компанії, які регулярно займаються організацією таких подій. Для них система дозволяє автоматизувати процеси реєстрації учасників, продажу квитків, управління подіями та моніторингу їхнього ходу. Наприклад, це можуть бути організатори конференцій для професіоналів у галузі ІТ, де важливо мати зручний інтерфейс для реєстрації учасників, продажу квитків і сповіщення про зміни в програмі. Також система буде корисною для організаторів культурних заходів, таких як концерти або виставки, де потрібно ефективно керувати продажем квитків і збором відгуків від відвідувачів. Крім того, система передбачає участь кінцевих користувачів, які зможуть легко реєструватися на заходи, купувати квитки та отримувати важливі повідомлення щодо події.

2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

Система створюється у вигляді веб-застосунку, що надає інструменти для організаторів подій з метою полегшення процесу реєстрації, управління квитками, сповіщення учасників, моніторингу перебігу заходу та аналітики після його завершення. Основна мета — створити безпечну та інтуїтивно зрозумілу систему, яка буде легко адаптуватися до різних типів заходів, таких як конференції, виставки, концерти, бізнес-зустрічі та тренінги.

Система реалізується у вигляді веб-додатку з сучасною клієнт-серверною архітектурою, що забезпечує розділення логіки між клієнтською та серверною частинами. Для серверної частини буде використано фреймворк Nest.js, який написаний на мові програмування TypeScript [4]. На боці клієнта буде застосовано фреймворк Next.js, який також використовує мову TypeScript. Завдяки SSR (Server Side Rendering) і зручному підходу до створення сторінок, Next.js забезпечить високу продуктивність та SEO-оптимізацію сайту [8].

В якості сховища даних запропоновано реляційну модель бази даних, що дозволить реалізувати надійне зберігання та структуровану обробку інформації в системі. Вона дозволяє надійно зберігати інформацію про користувачів, заходи, квитки, транзакції, повідомлення, відгуки та інші сутності. PostgreSQL обрано через її стабільність, підтримку складних запитів, транзакційність та можливість масштабування при зростанні обсягів даних [13].

Веб-застосунок буде хоститися на хмарній платформі AWS, що гарантує високу доступність, гнучкість у масштабуванні, стійкість до відмов та забезпечення безпеки. AWS дозволяє ефективно організувати деплой серверної та клієнтської частин, а також бази даних [15]. Вибір цієї платформи зумовлений її популярністю у промисловому використанні та широкими можливостями для моніторингу та автоматичного масштабування проєкту.

З метою забезпечення конфіденційності даних користувачів та захищеності системи будуть реалізовані заходи безпеки. Для безпечної передачі даних буде використано протокол TLS, що гарантує шифрування інформації між клієнтом та сервером. Також буде реалізовано механізми автентифікації та авторизації. Кожен користувач проходитиме перевірку особистості під час входу в систему, після чого отримає відповідні права доступу до функціоналу. Ролі в системі будуть поділені між адміністраторами, організаторами заходів та звичайними користувачами.

Серед ключових функціональних можливостей системи варто відзначити. Реєстрація та авторизація користувачів із можливістю входу через електронну пошту, номер телефону або сторонні сервіси. Створення, редагування та видалення подій організаторами, включаючи налаштування часу, місця проведення, опису, зображень і категорій. Система продажу квитків із прив'язкою до платіжних сервісів, генерацією унікального QR-коду для кожного квитка та перевіркою на вході. Інформування користувачів через email або push-сповіщення про майбутні події, зміни в розкладі або спеціальні пропозиції. Вивід аналітики для організаторів щодо кількості зареєстрованих учасників, динаміки продажів квитків, відгуків та рейтингу подій [17]. Система повинна витримувати навантаження, пов'язані зі збільшенням кількості користувачів, зберігати стабільну роботу при високих пікових навантаженнях та гарантувати безперервність обслуговування. Інтерфейс має бути доступним на різних пристроях, адаптивним та зрозумілим навіть для нових користувачів. Особлива увага приділяється збереженню даних — резервне копіювання, логування дій користувачів, захист від несанкціонованого доступу.

3 АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 UML проєктування ПЗ

Для забезпечення розуміння взаємодії користувачів із системою управління заходами було створено діаграму прецедентів. Інтерфейс системи надає різні рівні доступу, відображаючи доступні дії для звичайних користувачів та адміністраторів згідно їх прав. Дана діаграма є важливою складовою етапу проєктування, оскільки дозволяє візуалізувати вимоги до системи та взаємозв'язки між її елементами.



Рисунок 3.1 - Use case діаграма (рисунок виконаний самостійно)

Діаграма прецедентів візуалізує функціональні можливості системи керування заходами, моделюючи взаємодію акторів (користувача та адміністратора) з системою. Модель включає спеціалізовані залежності між прецедентами через зв'язки `<<include>>` та `<<extend>>`.

Система управління заходами надає користувачеві можливість переглядати доступні події, реєструватися на обрані з них, купувати квитки

для відвідування, а також залишати відгуки після їх завершення. Процес взаємодії користувача з системою починається з реєстрації та входу, що, як показує зв'язок <<include>>, невід'ємно включає перегляд доступних подій. Реєстрація на подію є опціональним розширенням перегляду подій, що дозволяє користувачеві перейти до наступного кроку після ознайомлення з інформацією. Аналогічно, купівля квитка може бути розширена отриманням сповіщення, ймовірно, про успішне придбання або деталі заходу [16].

Адміністратор системи має значно ширший спектр можливостей, спрямованих на керування всіма аспектами системи та забезпечення її належного функціонування. Він може створювати нові події, редагувати існуючі, видаляти застарілі або скасовані. Для кращого розуміння ефективності заходів та залучення користувачів, адміністратор може переглядати та аналізувати статистику подій. Крім того, до його обов'язків входить управління користувачами системи, а також надсилання їм важливих повідомлень та сповіщень. Для забезпечення контролю та оперативного реагування на будь-які нештатні ситуації, адміністратор здійснює моніторинг подій. Вся ця функціональність дозволяє адміністратору підтримувати актуальність інформації, ефективно взаємодіяти з користувачами та забезпечувати успішне проведення заходів.

3.2 Проєктування структури зберігання даних

Концептуальна модель даних у вигляді ER-діаграми була розроблена для візуалізації ключових об'єктів предметної області, їх властивостей та взаємозв'язків, що становить основу для реалізації надійного механізму зберігання та обробки інформації в системі керування івентами. База даних реалізується за допомогою системи керування базами даних PostgreSQL, яка є надійною, масштабованою та підтримує всі необхідні функції для роботи з реляційними структурами. Нижче представлена ER-діаграма, яка відображає логічну модель структури даних, що буде використовуватись у проєкті.

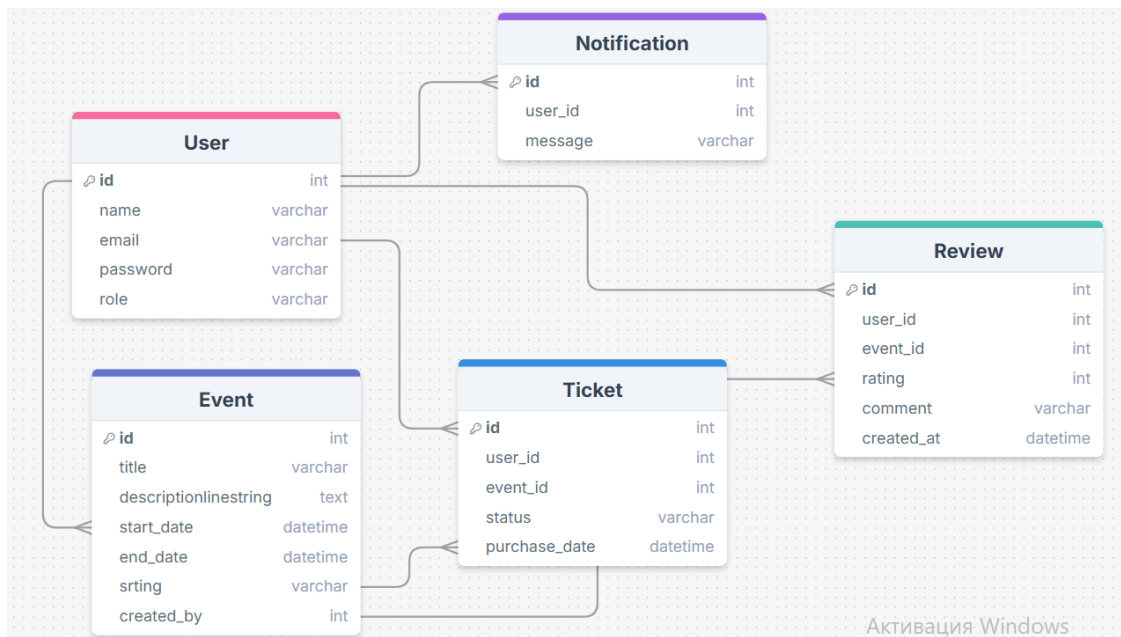


Рисунок 3.2 – Проектування структури зберігання даних (рисунок виконаний самостійно)

Ця ER-діаграма ілюструє структуру бази даних для системи управління заходами, яка дозволяє організувати події, реєструвати учасників, продавати квитки, отримувати відгуки та надсилати сповіщення. Основні сутності в базі даних — це користувач, подія, квиток, відгук та сповіщення.

Користувач є центральною сутністю в системі. Кожен користувач має ідентифікатор, ім'я, електронну пошту, зашифрований пароль та роль, що дозволяє визначати його функціональні можливості в системі (наприклад, організатор чи учасник подій). Користувач може створювати події, купувати квитки для заходів, писати відгуки після відвідування заходів та отримувати сповіщення від системи.

Подія описує конкретний захід. Вона має також унікальний ідентифікатор, назву, опис, дату початку та завершення та місце проведення. Кожна подія також містить ідентифікатор користувача, який її створив. Це дозволяє зберігати інформацію про організаторів заходів. Події можуть мати

кілька квитків та відгуків, що дозволяє відстежувати кількість учасників і їхні враження від заходу. Квиток є обов'язковим елементом для участі в події. Кожен квиток містить ідентифікатор, ідентифікатор користувача, який його купив, ідентифікатор події, на яку квиток куплений, статус — тобто, використаний чи не використаний, а також дату покупки. Це дозволяє ефективно відслідковувати всі покупки та їхній статус. Відгук дозволяє користувачам оцінювати події після їх відвідування. Відгук включає в себе оцінку, коментар і дату створення. Кожен відгук прив'язаний до конкретного користувача та події, що дозволяє зберігати історію відгуків та забезпечує прозорість у взаємодії користувачів з заходами.

Сповіщення є важливим елементом для підтримки взаємодії з користувачами. Вони можуть включати важливі повідомлення про зміни в події або нагадування про захід. Сповіщення містять ідентифікатор користувача, який отримує повідомлення, текст повідомлення, прапорець прочитаності та дату надсилання. Сповіщення допомагають організаторам заходів тримати користувачів в курсі важливих оновлень і покращують взаємодію з ними.

Зв'язки між сутностями вказують на важливі аспекти функціональності системи. Користувач може створювати кілька подій, купувати кілька квитків, писати кілька відгуків та отримувати кілька сповіщень. Подія може мати кілька квитків та відгуків. Ці зв'язки дозволяють чітко відстежувати всі взаємодії в системі, забезпечуючи її ефективність і простоту у використанні.

3.3 Приклади найцікавіших алгоритмів та методів

З метою кращого розуміння структури обміну інформацією між користувачами, окремими системами та базами даних у рамках розробленого програмного рішення реалізовано, було побудовано діаграму потоків даних. Ця діаграма відображає загальну взаємодію між користувачем,

адміністратором, основними сервісами системи — такими як управління подіями, продаж квитків, обробка відгуків та автентифікація, а також базами даних. Побудова такої моделі дозволяє представити логіку функціонування системи та взаємозв'язок її компонентів [12].

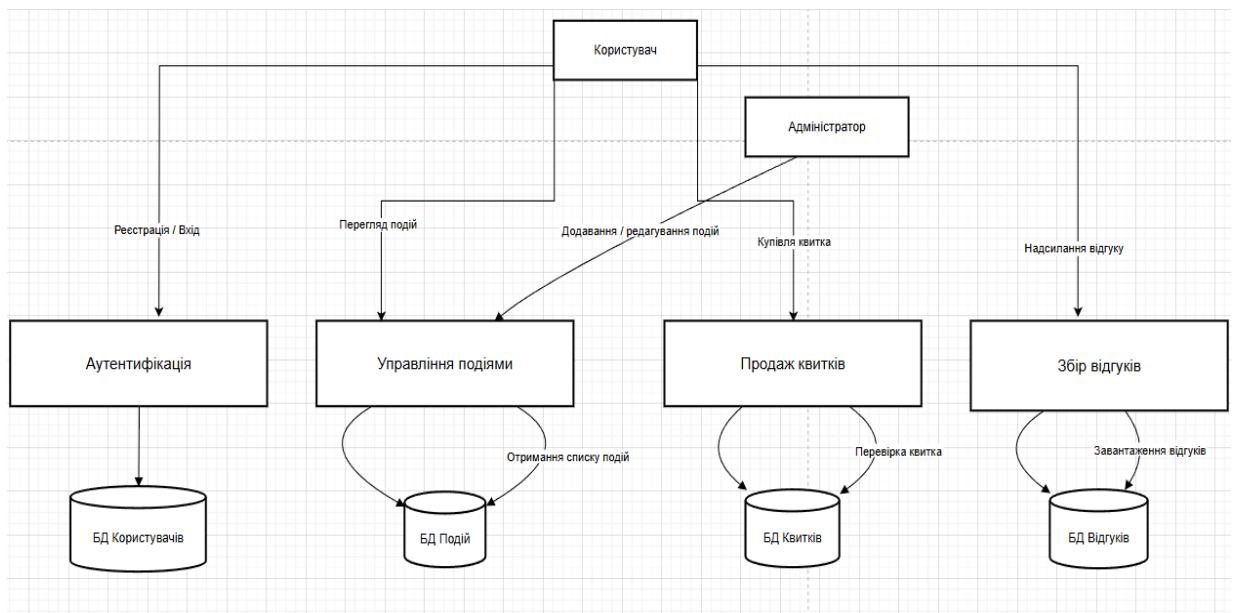


Рисунок 3.3 - Data Flow Diagram (рисунок виконаний самостійно)

Діаграма демонструє, як дані передаються від користувачів до системи, обробляються в межах відповідних модулів, зберігаються у базах даних та виводяться у вигляді інформації. Кожен процес у системі чітко взаємодіє зі своєю частиною даних, що дозволяє мінімізувати можливі конфлікти та помилки при роботі з інформацією. Така структура забезпечує логічний та послідовний потік даних, що сприяє кращій організації бізнес-логіки та підтримці внутрішньої цілісності системи. Вона також полегшує масштабування проєкту в майбутньому, оскільки окремі модулі можуть бути доопрацьовані або замінені без порушення загальної архітектури.

3.4 Створення UI/UX

Одним з ключових етапів розробки веб-застосунку є створення зручного та інтуїтивно зрозумілого інтерфейсу користувача та забезпечення позитивного досвіду користувача. На цьому етапі проектуються та розробляються всі візуальні елементи сторінок, їхня структура та інтерактивність.

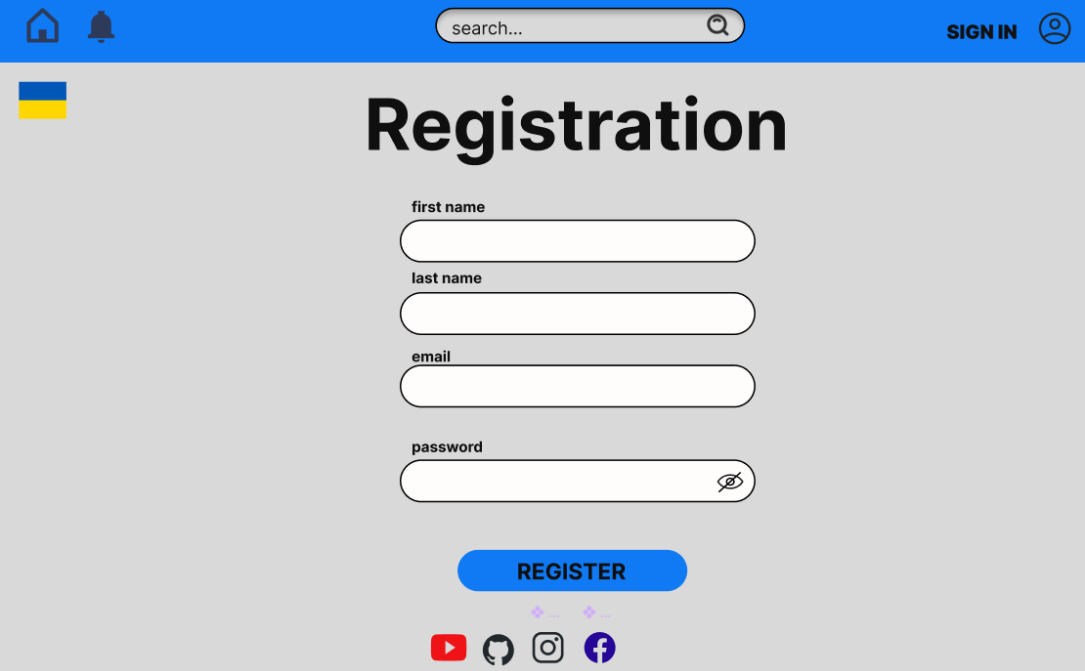
The image shows a web registration form. At the top, there is a blue navigation bar containing a home icon, a notification bell, a search bar with the text "search...", and a "SIGN IN" button with a user profile icon. Below the navigation bar is a grey main content area. On the left side of this area is a small Ukrainian flag. The main heading is "Registration" in a large, bold, black font. Below the heading are four input fields: "first name", "last name", "email", and "password". The "password" field includes an eye icon for toggling visibility. A prominent blue "REGISTER" button is centered below the form fields. At the bottom of the registration area, there are social media icons for YouTube, Telegram, Instagram, and Facebook.

Рисунок 3.4 – Сторінка реєстрації користувача (рисунок виконаний самостійно)

На зображенні показано інтерфейс сторінки реєстрації користувача веб-сайту, виконаний у мінімалістичному стилі з контрастними кольорами: синій для заголовка та кнопки, білий для полів введення та тексту. Угорі розміщено заголовок "REGISTERS", нижче — чотири поля для введення імені, прізвища, електронної пошти та пароля (з іконкою приховування символів). Під ними знаходиться кнопка "REGISTER".

Наступним важливим кроком є створення сторінки для входу в існуючий обліковий запис. Ця сторінка має бути простою, дозволяючи користувачам швидко отримати доступ до функціоналу веб-застосунку.

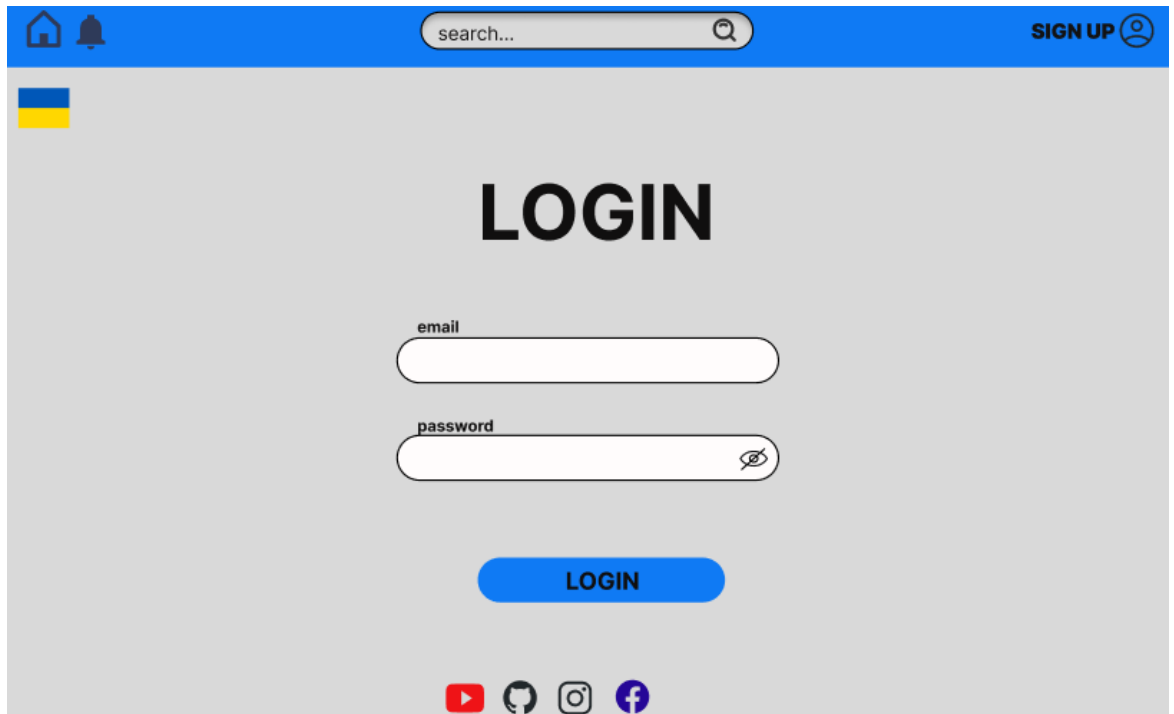


Рисунок 3.5 – Сторінка входу користувача (рисунок виконаний самостійно)

Сторінка входу користувача має простий та інтуїтивно зрозумілий інтерфейс. У центрі екрану розміщені два основні поля для введення даних: електронної пошти та пароля. Поле введення пароля доповнене іконкою приховування/відображення символів, що дозволяє користувачу переконатися в правильності введеного значення. Під полями розташована кнопка "Увійти", яка виконує авторизацію користувача у системі після перевірки введених даних. Дизайн витриманий у тому ж стилі, що й інші сторінки сайту.

Після успішної реєстрації або входу в обліковий запис користувачу відображається головна сторінка веб-застосунку. Головна сторінка слугує центральним хабом, надаючи огляд доступних подій та пропонуючи швидкий доступ до ключових функцій.

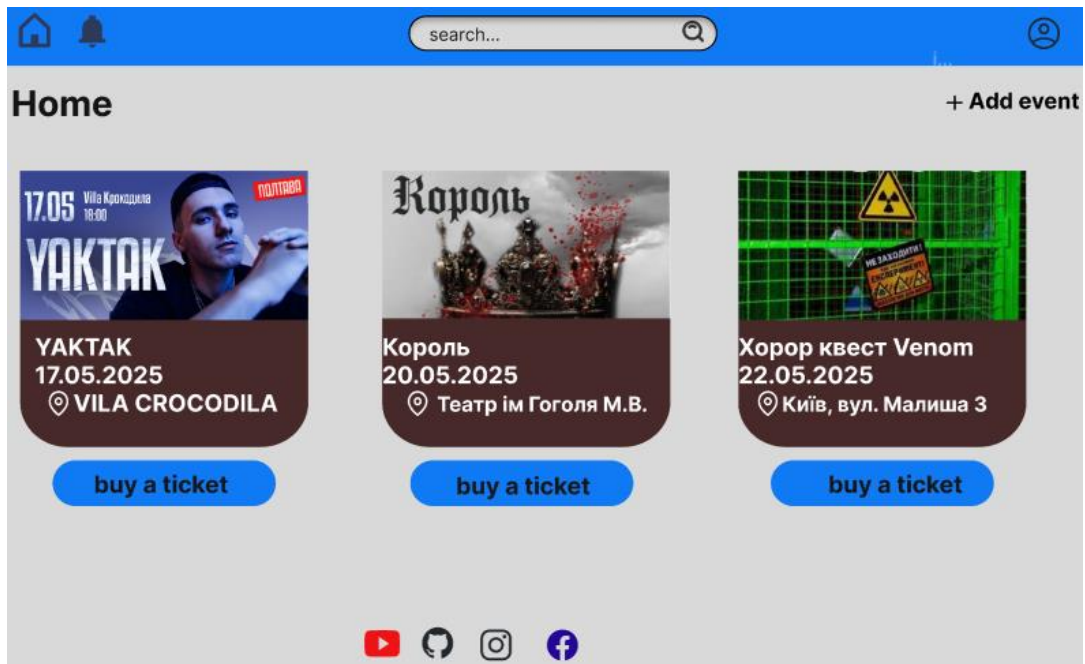


Рисунок 3.6 – Головна сторінка (рисунок виконаний самостійно)

Головна сторінка веб-застосунку відображає список майбутніх подій у вигляді окремих карток, що містять назву події, дату, місце проведення та короткий опис. Кожна картка також має кнопку "Купити квиток", яка дозволяє швидко перейти до оформлення замовлення.

Для створення нової події в системі користувачу потрібно вибрати кнопку "Add event" на головній сторінці. Після натискання на неї відкривається спеціальна форма, де користувач може ввести всю необхідну інформацію про подію. Вона дозволяє додати назву події, опис, час проведення, місце, а також встановити вартість квитків.

The screenshot shows a mobile application interface for adding an event. At the top, there is a blue header with a home icon, a notification bell, a search bar with the text "search...", and a user profile icon. Below the header, the main content area is titled "Add event". On the left side, there is a large grey square with a white document icon containing a plus sign, with the text "Add photo" below it. On the right side, there are several input fields: "Name" (a rounded rectangle), "Description" (a larger rounded rectangle), "Location" (a rounded rectangle with a dropdown arrow), and "Price" (a rounded rectangle). Above the "Location" field, there are two date pickers, each consisting of a calendar icon and a rounded rectangle. At the bottom center, there is a prominent blue rounded rectangle button with the text "create" in white.

Рисунок 3.7 – Сторінка додавання події (рисунок виконаний самостійно)

Окрім цього, є можливість додавати фотографію, що дозволить користувачам візуально представити захід. Після заповнення всіх полів, користувач може натискати кнопку "Створити", щоб зберегти нову подію в системі та зробити її доступною для інших користувачів.

Після вибору конкретної події на головній сторінці користувачу відкривається сторінка з детальною інформацією про неї. Ця сторінка надає вичерпні дані про захід, включаючи його опис, місце проведення, доступні дати та час, а також можливість придбання квитків.

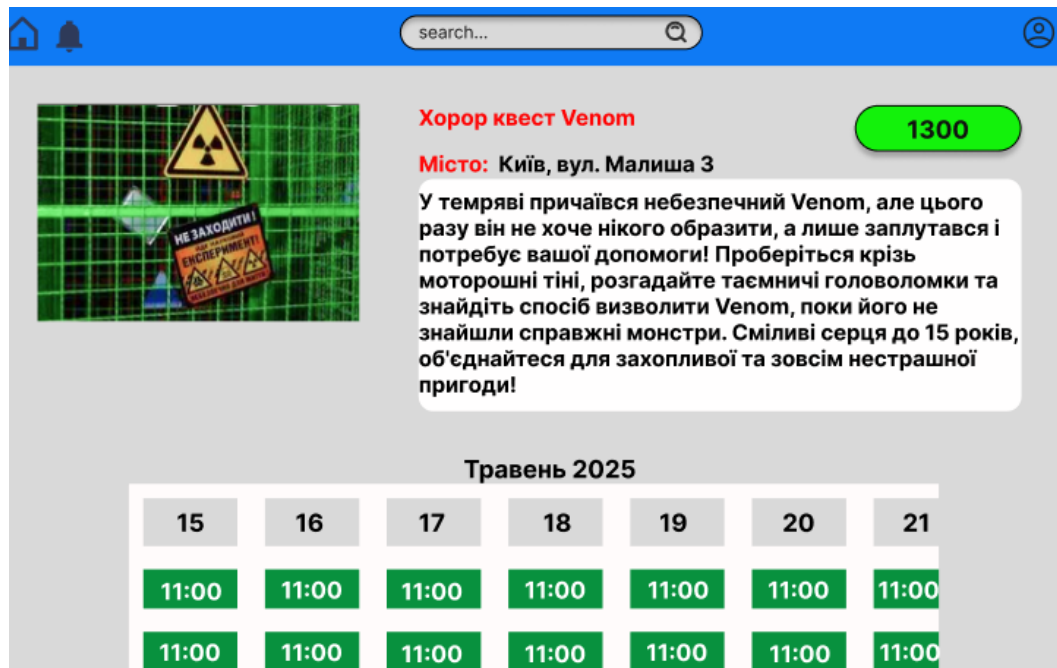


Рисунок 3.8 – Сторінка перегляду події (рисунок виконаний самостійно)

Сторінка перегляду детальної інформації про хорор-квест "Venom" надає всю необхідну інформацію про подію. Вона містить велике фото, що відображає атмосферу квесту. Під зображенням вказано назву — "Хорор квест Venom", що підкреслює жанр заходу. Також зазначено місце проведення — "Київ, вул. Малиша 3", і є опис події, який пояснює, що квест призначений для дітей до 15 років. Ціна квитка — 1300 грн. Окрім цього, на сторінці присутній інтерактивний календар на травень 2025 року, де можна вибрати доступні дати та час сеансів. У верхній частині сторінки є елементи навігації для повернення на головну, доступу до профілю та перегляду інших подій, що робить користування сайтом зручним і інтуїтивно зрозумілим.

Після вибору бажаної дати та часу сеансу на сторінці перегляду події, користувачу відображається сторінка оформлення покупки квитка. На цій сторінці користувач підтверджує обрані параметри та вводить необхідну інформацію для завершення транзакції.

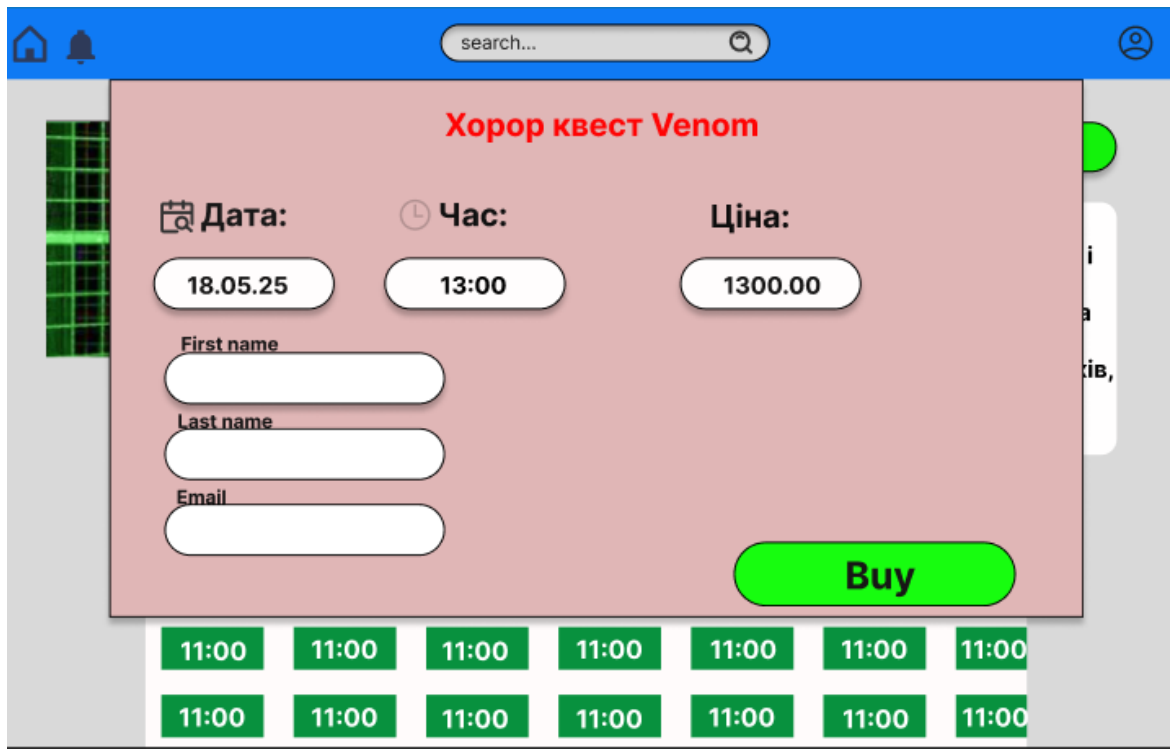


Рисунок 3.9 – Вікно купівлі білету (рисунок виконаний самостійно)

Спливаюче вікно оформлення покупки квитка на хорор-квест "Venom" надає користувачам можливість швидко завершити покупку. У вікні відображається назва події "Хорор квест Venom", обрана дата є "18.05.25" та час "13:00", а також ціна квитка — "1300.00 грн". Користувачам пропонується заповнити поля для введення імені, прізвища та електронної пошти. Після заповнення необхідних полів, користувач може натискати на велику зелену кнопку "Купити", щоб завершити оформлення покупки квитка.

Після успішного завершення процесу покупки квитка, користувач отримує електронний квиток на вказану ним електронну пошту. Цей квиток містить всю необхідну інформацію про подію, яку можна пред'явити для входу.



Рисунок 3.10 – Квиток на подію (рисунок виконаний самостійно)

Електронний квиток на хорор-квест "Venom" містить всю необхідну інформацію для участі у події. На квитку зазначено його унікальний номер — "Квиток №999", а також назва події — "Хорор квест Venom". Інформація про місце проведення події вказана як "Місто: Київ, вул. Малиша 3". В правому нижньому куті квитка розташовані логотипи соціальних мереж, що дозволяють користувачам взаємодіяти з офіційними акаунтами заходу в мережі.

4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

У цьому розділі детально розглядаються всі програмні рішення, прийняті в процесі розробки веб-системи для управління заходами. Від якості цих рішень залежить стабільність, продуктивність і масштабованість системи, а також зручність її використання як для звичайних користувачів, так і для адміністраторів. Ретельно підібраний стек технологій і продумана архітектура є основою успішного функціонування програмного продукту [10].

Основною метою цієї розробки було створення сучасної веб-платформи, яка є інтуїтивно зрозумілою у використанні, надійною з точки зору захисту та доступності даних, а також легко масштабується для задоволення зростаючих вимог і зростаючої кількості користувачів [7]. Особливу увагу було приділено тому, щоб система могла адаптуватися до змін і легко розширюватися новими функціональними модулями за потреби.

На стороні сервера додаток створено з використанням фреймворку Nest.js у поєднанні з TypeScript, що дозволило розробити надійну та структуровану архітектуру серверної частини. Для зберігання даних використовувалася реляційна база даних PostgreSQL завдяки її високій продуктивності, потужним можливостям запитів і підтримці складних зв'язків між таблицями [18]. На стороні клієнта система використовує фреймворк Next.js, також заснований на TypeScript, який забезпечує швидкий і динамічний інтерфейс. Цей вибір робить зовнішній інтерфейс дуже придатним для обслуговування та ефективним, особливо для відтворення статичного та серверного вмісту. Інтерфейс користувача розроблено з урахуванням зручності використання, щоб забезпечити постійну роботу як для адміністраторів, так і для звичайних користувачів. Було впроваджено гнучкі принципи розміщення, щоб зробити систему доступною на різних пристроях, включаючи настільні ПК, планшети та мобільні телефони.

При розробці та впровадженні програмної системи використовувався набір сучасних технологій — технологій, які широко поширені у сфері веб-розробки та зарекомендували себе як ефективні та стабільні.

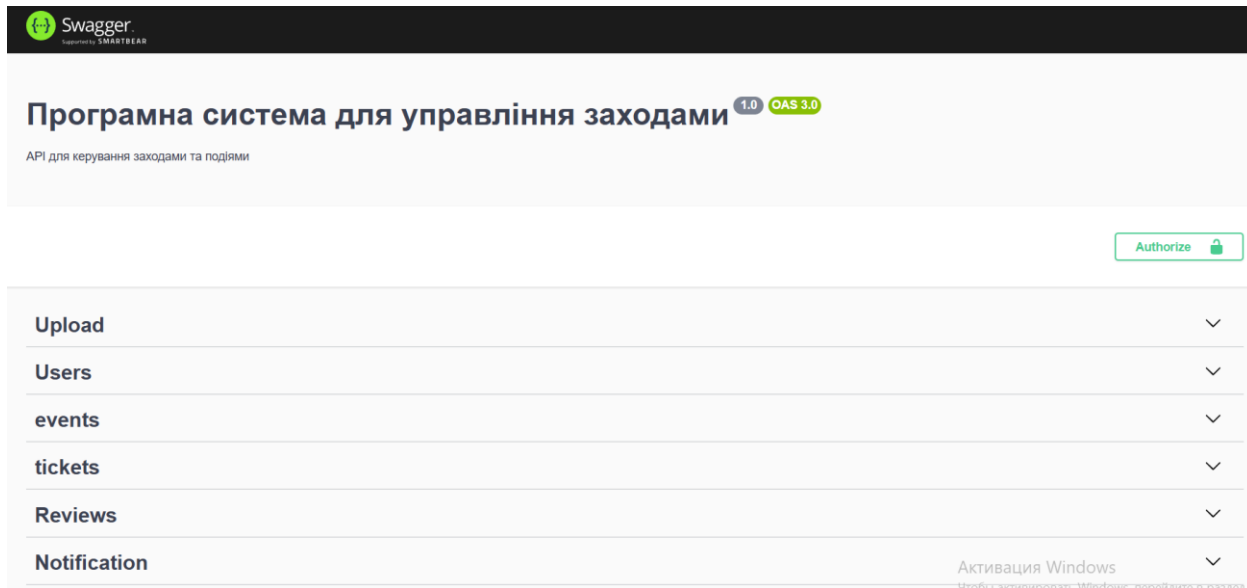


Рисунок 4.1 – Головна сторінка Swagger UI (рисунок виконаний самостійно)

Головна сторінка програми на стороні клієнта представляє зрозумілий і доступний макет, що дозволяє користувачам легко переглядати майбутні події, переглядати деталі та купувати квитки. Нижче наведено ілюстрацію інтерфейсу домашньої сторінки, яку бачить типовий відвідувач.

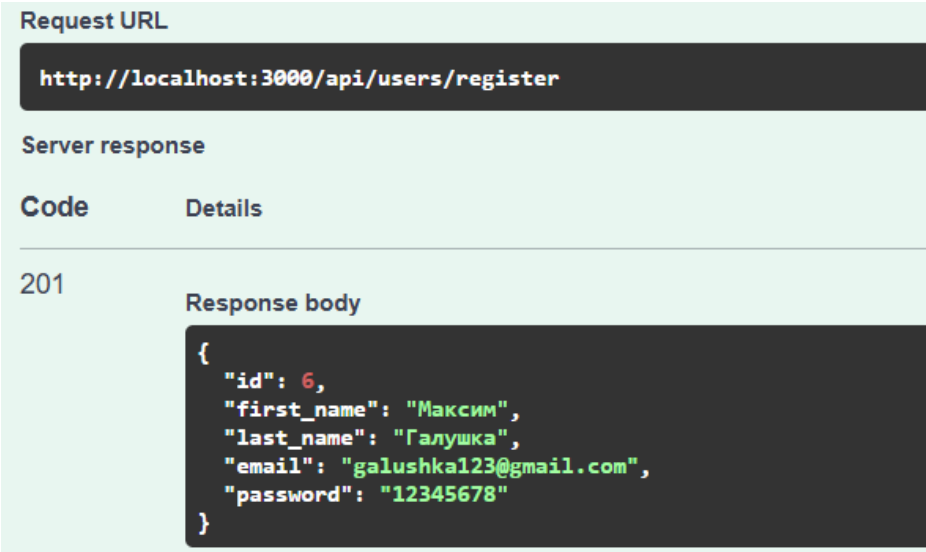


Welcome to Event Management System

Please sign in or register to access all features

Рисунок 4.2 – Головна сторінка клієнтської частини (рисунок виконаний самостійно)

Одним із головних завдань було створення механізму для реєстрації, входу в систему та керування користувачами. Для перевірки роботи всіх частин системи та зручності тестування використовувався спеціальний інтерфейс, у якому можна переглядати доступні функції системи, надсилати запити та бачити відповіді. Система надає можливості реєстрації нових користувачів, автентифікації за обліковими даними, перегляду списку зареєстрованих користувачів та їх видалення за унікальним ідентифікатором. Такий метод значно зменшує розробку, перевірку та налаштування роботи програми. Пріоритетами розробки були: захист конфіденційності користувачів, інтуїтивність інтерфейсу та гарантована цілісність даних.



The screenshot displays an API client interface with the following details:

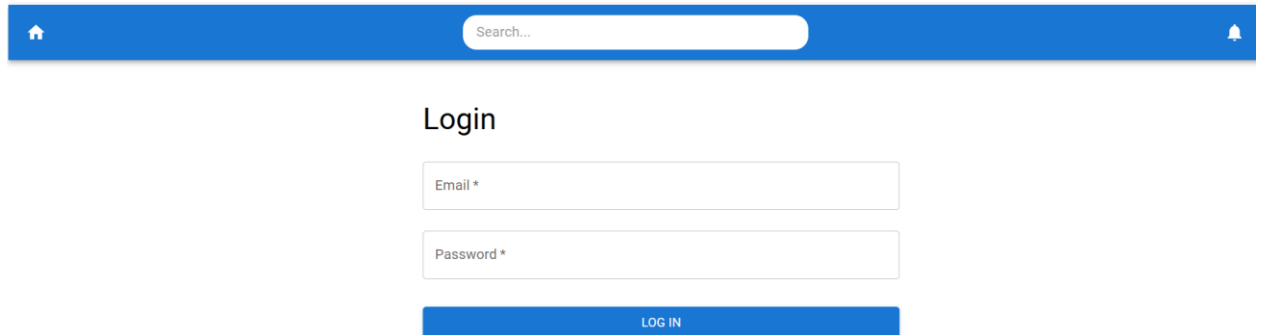
- Request URL:** `http://localhost:3000/api/users/register`
- Server response:**

Code	Details
201	<p>Response body</p> <pre>{ "id": 6, "first_name": "Максим", "last_name": "Галушка", "email": "galushka123@gmail.com", "password": "12345678" }</pre>

Рисунок 4.3 – Результат створення користувача (рисунок виконаний самостійно)

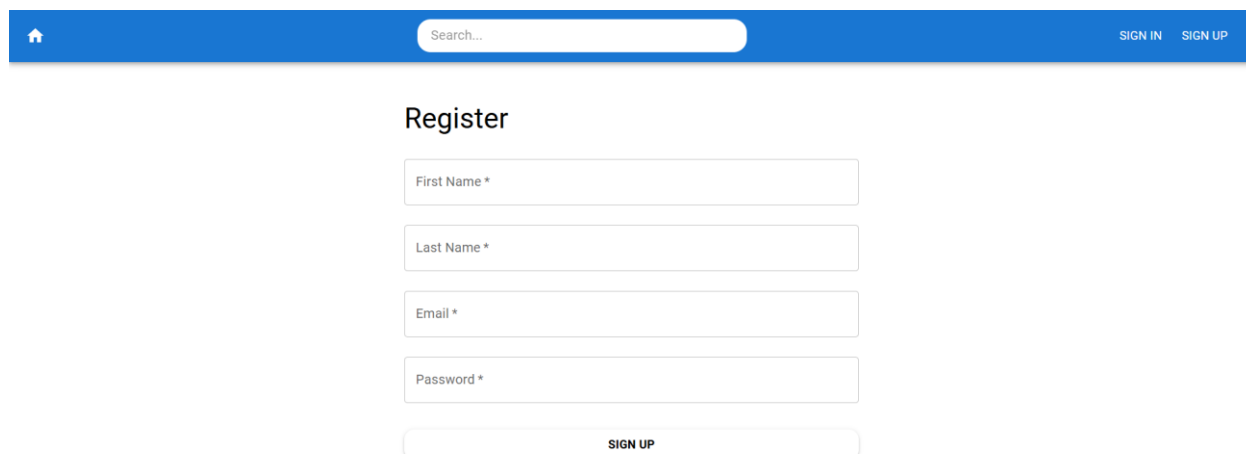
З боку клієнта також реалізовано зручні форми для реєстрації та входу в систему. Інтерфейс побудований таким чином, щоб користувач міг легко створити обліковий запис або увійти до свого профілю, не витрачаючи зайвого

часу. Нижче наведено приклади зовнішнього вигляду форм реєстрації та авторизації на клієнтській частині платформи.



The screenshot shows a user login interface. At the top, there is a blue navigation bar containing a home icon on the left, a search bar in the center, and a notification bell icon on the right. Below the navigation bar, the page title 'Login' is centered. Underneath the title, there are two white input fields: the first is labeled 'Email *' and the second is labeled 'Password *'. At the bottom of the form, there is a solid blue button with the text 'LOG IN' in white capital letters.

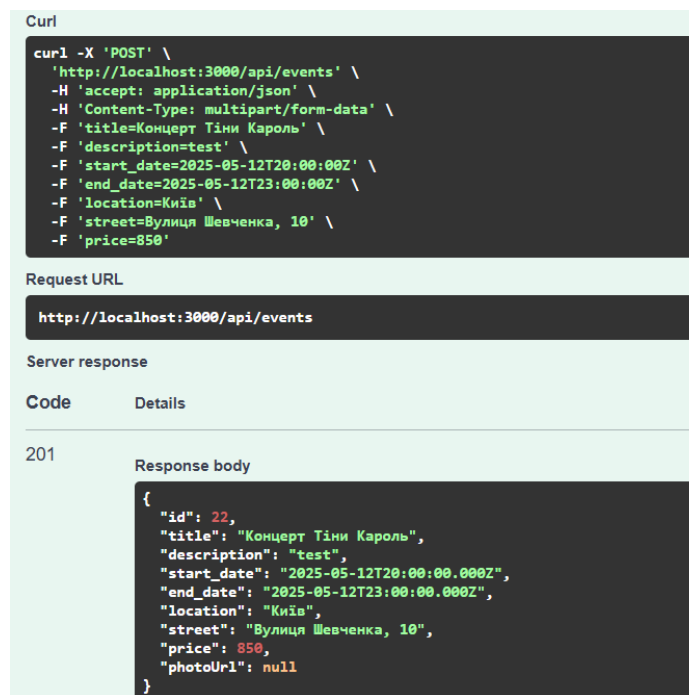
Рисунок 4.4 – Сторінка авторизації користувача (рисунок виконаний самостійно)



The screenshot shows a user registration interface. At the top, there is a blue navigation bar containing a home icon on the left, a search bar in the center, and 'SIGN IN' and 'SIGN UP' links on the right. Below the navigation bar, the page title 'Register' is centered. Underneath the title, there are four white input fields: the first is labeled 'First Name *', the second is labeled 'Last Name *', the third is labeled 'Email *', and the fourth is labeled 'Password *'. At the bottom of the form, there is a white button with the text 'SIGN UP' in blue capital letters.

Рисунок 4.5 – Сторінка реєстрації користувача (рисунок виконаний самостійно)

Для забезпечення повноцінного функціонування системи управління заходами було розроблено набір API-ендпоінтів, що реалізують можливість керування подіями. Цей функціонал надає адміністраторам змогу створювати, переглядати, редагувати та видаляти події, забезпечуючи гнучкий контроль над контентом, що відображається на платформі. Взаємодія з цими точками доступу може здійснюватися як через користувацький інтерфейс, так і за допомогою інструментів для розробників, таких як Swagger UI, що значно спрощує процес тестування та інтеграції. API для керування подіями включає низку основних операцій. Зокрема, ендпоінт для отримання списку всіх подій дозволяє завантажити повний перелік наявних заходів, які можуть відображатися на головній сторінці веб-сайту. Операція створення події дає змогу адміністраторам додавати нові заходи до системи, вказуючи при цьому всі необхідні параметри: назву, опис, дату та час проведення, місце, вартість та інші деталі. Редагування подій здійснюється через відповідний ендпоінт, який дозволяє вносити зміни до раніше створених записів, зокрема оновлювати статус, змінювати текстову інформацію або редагувати вартість.



```
Curl
curl -X 'POST' \
'http://localhost:3000/api/events' \
-H 'accept: application/json' \
-H 'Content-Type: multipart/form-data' \
-F 'title=Концерт Тіни Кароль' \
-F 'description=test' \
-F 'start_date=2025-05-12T20:00:00Z' \
-F 'end_date=2025-05-12T23:00:00Z' \
-F 'location=Київ' \
-F 'street=Вулиця Шевченка, 10' \
-F 'price=850'

Request URL
http://localhost:3000/api/events

Server response
Code      Details
201
Response body
{
  "id": 22,
  "title": "Концерт Тіни Кароль",
  "description": "test",
  "start_date": "2025-05-12T20:00:00.000Z",
  "end_date": "2025-05-12T23:00:00.000Z",
  "location": "Київ",
  "street": "Вулиця Шевченка, 10",
  "price": 850,
  "photoUrl": null
}
```

Рисунок 4.6 – Результат створення заходу (рисунок виконаний самостійно)

У наведеному прикладі створюється подія під назвою "Концерт Тіни Кароль" з описом "test". Подія відбувається за адресою "Київ, Вулиця Шевченка, 10", має вказані дату та час проведення, а також ціну — 850 гривень. Цей приклад наочно демонструє, як саме відбувається процес створення події через API [5].

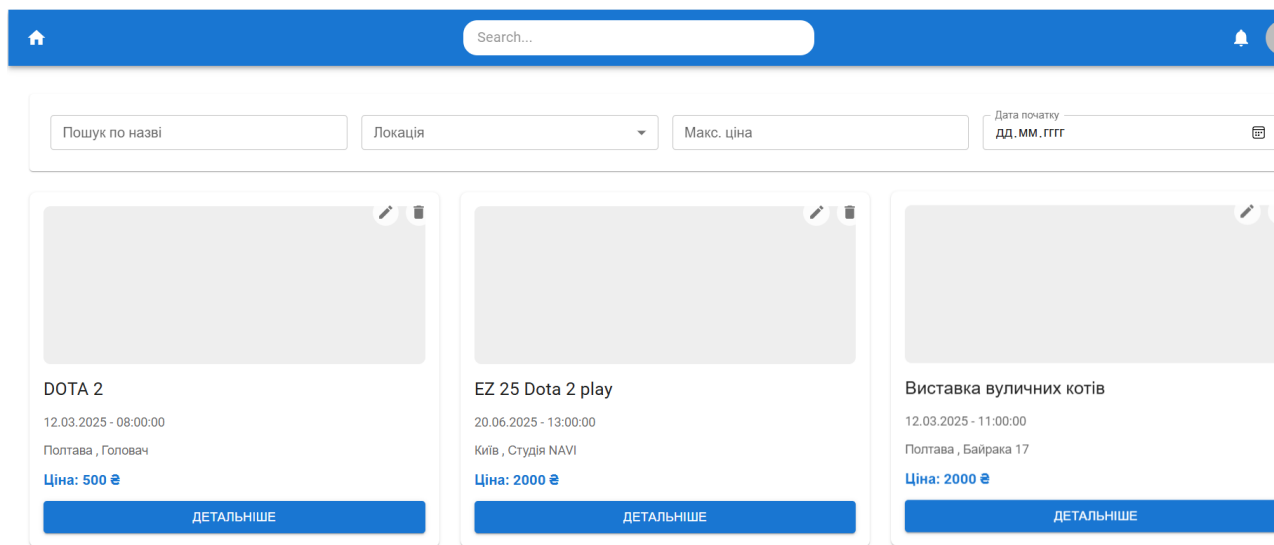


Рисунок 4.7 – Сторінка з заходами (рисунок виконаний самостійно)

На основі реалізованого API було створено зручний інтерфейс на клієнтській частині платформи, де користувачі можуть переглядати всі доступні події, ознайомлюватися з їхніми деталями та купувати квитки.

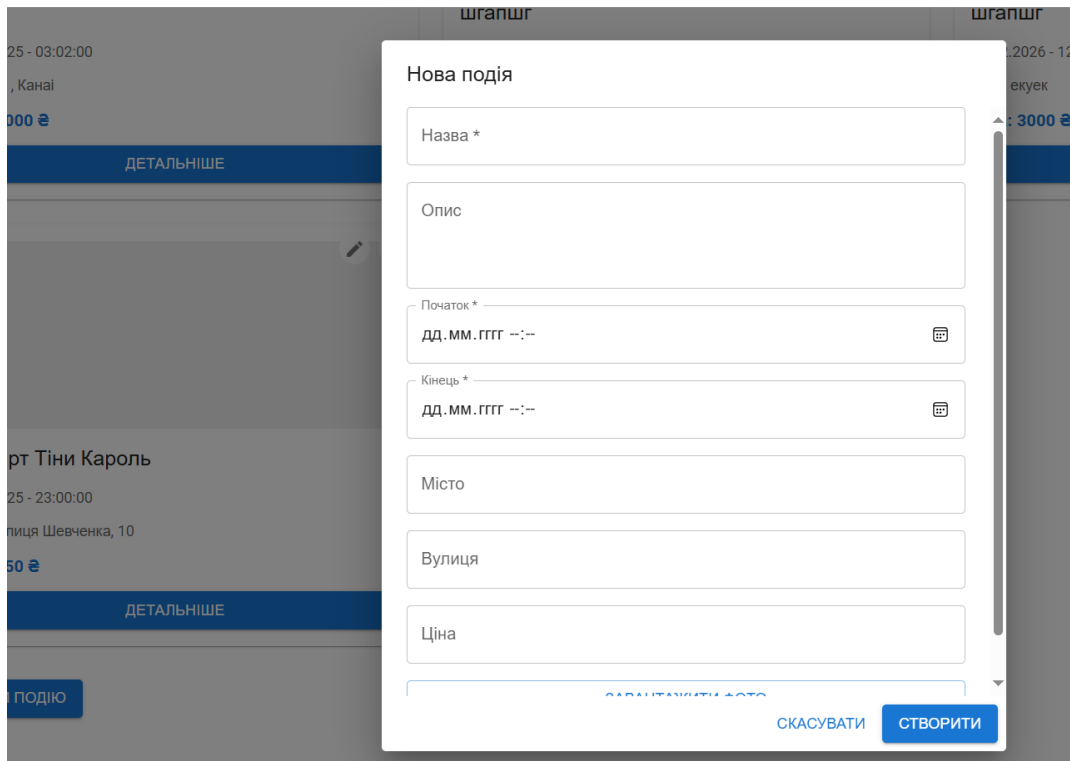


Рисунок 4.8 – Вікно додавання події (рисунок виконаний самостійно)

У системі управління заходами реалізована функція сповіщень, яка дозволяє адміністраторам надсилати повідомлення користувачам. Це потрібно для того, щоб інформувати людей про важливі події, зміни або нагадування, пов'язані з їхніми квитками чи участю в заходах.

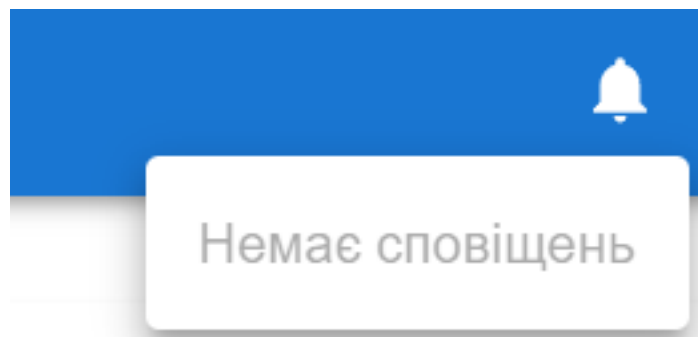


Рисунок 4.9 – Система сповіщення (рисунок виконаний самостійно)

```
curl -X 'POST' \
  'http://localhost:3000/api/notifications' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "user_id": 6,
    "message": "Приємного вечора, ваш квиток №76"
  }'
```

Request URL

```
http://localhost:3000/api/notifications
```

Server response

Code	Details
201	<p>Response body</p> <pre>{ "id": 1, "user_id": 6, "message": "Приємного вечора, ваш квиток №76" }</pre>

Рисунок 4.10 – Результат надсилання сповіщення (рисунок виконаний самостійно)

Адміністратор може надіслати персональне повідомлення конкретному користувачу. Наприклад, як показано на зображенні вище, де вказується ID користувача і текст повідомлення — у цьому випадку: "Приємного вечора, ваш квиток №76". У відповідь сервер надсилає підтвердження, що повідомлення успішно створене, разом з його ID, даними користувача та текстом. Крім окремих повідомлень, система також підтримує можливість розсилки повідомлень одразу багатьом користувачам. Це дуже зручно, коли потрібно повідомити про початок або скасування події, зміну часу, появу нових квитків, акції або будь-які інші новини. Повідомлення можуть приходити в інтерфейсі сайту.

Наступним кроком є придбання квитка на подію. У системі є зручна функція, яка дозволяє купити квиток на певну подію. Щоб це зробити, користувач просто заповнює невелику форму, де вказує своє ім'я, прізвище, електронну адресу та вибирає подію, на яку хоче піти.

```

Curl
curl -X 'POST' \
'http://localhost:3000/api/tickets' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "first_name": "Максим",
  "last_name": "Галушка",
  "email": "galushka123@gmail.com",
  "event_id": 14
}'

Request URL
http://localhost:3000/api/tickets

Server response
Code      Details
-----
201      Response body
{
  "id": 7,
  "user_id": null,
  "event_id": 14,
  "status": "pending",
  "purchase_date": "2025-05-31T10:04:58.071Z",
  "first_name": "Максим",
  "last_name": "Галушка",
  "email": "galushka123@gmail.com"
}

```

Рисунок 4.11 – Результат купівлі квитка (рисунок виконаний самостійно)

На зображенні вище показано приклад, як такий квиток створюється через запит до серверу. У запиті зазначено ім'я — "Максим", прізвище — "Галушка", e-mail та ID події — 14. Після цього система створює новий квиток і надсилає відповідь із підтвердженням, що все пройшло успішно. У відповіді вказано ID квитка, на яку його купили, дату покупки, а також статус — "pending", тобто квиток ще не підтверджено [6].

На клієнтській частині приклад цього процесу показано нижче — інтерфейс подачі заявки на квиток є зрозумілим.

Виставка вуличних котів ×

Опис
Виставка Котів Дорогі кияни та гості Києва! Ласкаво просимо Вас відвідати нашу зимову виставку котів та ярмарок зоотоварів! На виставці Ви зможете познайомитись з різними породами котів, придбати корми та аксесуари для своїх домашніх улюбленців, а також відпочити у нашому смачному дворіку! Подаруйте собі та своїм дітям яскраве зимове свято, завітайте до нашої виставки!

Деталі
Дата та час:
12.03.2025, 11:00:00 – 12.03.2025, 15:00:00
Місце проведення:
Полтава, Байрака 17
Ціна:
2000 ₴

Відгуки

Купити квиток

Ім'я *

Прізвище *

Email *

Номер картки *

Термін дії * CVV *

ОПЛАТИТИ

Активаци
Чтобы актив
"Параметр

Рисунок 4.12 – Вікно купівлі квитка (рисунок виконаний самостійно)

Клієнтський інтерфейс також включає зручну систему фільтрації подій для покращення взаємодії з користувачем. Користувачі можуть легко шукати події за назвою, місцезнаходженням, діапазоном цін або датою. Ця функція дозволяє відвідувачам швидко знаходити події, які відповідають їхнім інтересам і наявності, що робить процес перегляду більш ефективним і зручним для користувача.

Пошук по назві

Локація

Макс. ціна

Дата початку
ДД.ММ.ГГГГ

Рисунок 4.13 – Фільтрація пошуку (рисунок виконаний самостійно)

5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тестування веб-застосунку для управління заходами є завершальним етапом розробки, що дозволяє переконатися в правильності роботи основних функцій системи. Під час тестування особливу увагу було приділено перевірці процесів реєстрації подій, реєстрації користувачів, авторизації, а також взаємодії з подіями. Мета тестування — забезпечити стабільну роботу інтерфейсу та коректну обробку користувацьких дій у реальному середовищі. У цьому розділі наведено тест-кейси, які демонструють ефективність реалізованої логіки та зручність використання системи.

Таблиця 5.1 – Тест-кейс №1 (таблиця виконана самостійно)

Інформація про тест-кейс			
Ідентифікатор тесту:	Тест-кейс №1		
Опис функції:	Створення нового заходу користувачем		
Власник тесту:	Галушка Максим Віталійович		
Дата створення:	28.05.2025		
Мета тесту:	Перевірити правильність функції створення заходу авторизованим користувачем		
Передумова			
№	Опис випадку	Очікуваний результат	Висновок
1	Відкрити веб-застосунок	Користувач має доступ до сайту, який відкритий	Пройдено
2	Авторизуватися через логін та пароль	Користувача перенаправлено на головну сторінку	Пройдено

Створення нового заходу			
№	Опис випадку	Очікуваний результат	Висновок
1	Натиснути кнопку «Створити захід»	Відкривається форма для заповнення	Пройдено
2	Ввести назву заходу	Назва зберігається, перевірка обмеження по символах	Пройдено
3	Ввести опис заходу	Поле «Опис» заповнено та перевірено на допустиму кількість символів	Пройдено
4	Обрати дату та час проведення	Обрано дату/час, перевірено на валідність	Пройдено
5	Вказати вартість участі в заході	Поле «Ціна» заповнено коректно	Пройдено
6	Ввести назву міста та адресу проведення	Адреса збережена в полі «Локація»	Пройдено
7	Натиснути кнопку «Зберегти»	Подію успішно створено, з'являється повідомлення про успішне збереження	Пройдено
8	Перейти до перегляду створеної	Відкривається сторінка події з усіма	Пройдено

	події через кнопку «Докладніше»	вказаними раніше даними	
9	Перевірити правильність відображення інформації	Назва, опис, дата, час, ціна та локація відповідають введеним даним	Пройдено
Результати тестування			
Тестувальник: Галушка М.В.		Дата прогону тесту: 28.05.2025	Результат тесту (P/F/V): ПРОЙДЕНО (P)

Таблиця 5.2 – Тест-кейс №2 (таблиця виконана самостійно)

Інформація про тест-кейс			
Ідентифікатор тесту:		Тест-кейс №2	
Опис функції:		Придбання квитка на захід	
Власник тесту:		Галушка Максим Віталійович	
Дата створення:		29.05.2025	
Мета тесту:		Перевірити, чи може авторизований користувач успішно придбати квиток на захід через веб-застосунок	
Передумова			
№	Опис випадку	Очікуваний результат	Висновок
1	Користувач має обліковий запис	Дані для входу є у користувача	Пройдено
Створення нового заходу			

№	Опис випадку	Очікуваний результат	Висновок
1	Увійти в систему	Відображається головна сторінка	Пройдено
2	Перейти на головну сторінку заходів	Відображається список доступних заходів	Пройдено
3	Вибрати захід зі списку	Відкривається сторінка з деталями події	Пройдено
4	Натиснути «Детальніше»	Відкривається форма покупки квитка	Пройдено
5	Заповнити поля: ім'я, прізвище, електронна пошта, номер картки	Усі поля заповнені коректно	Пройдено
6	Підтвердити оплату	З'являється повідомлення про успішну оплату	Пройдено
7	З'являється повідомлення про успішну оплату	Придбаний квиток присутній у списку	Пройдено
Результати тестування			
Тестувальник: Галушка М.В.		Дата прогону тесту: 29.05.2025	Результат тесту (P/F/B): ПРОЙДЕНО (P)

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було здобуто цінний практичний досвід розробки повнофункціонального веб-застосунку з використанням сучасних технологій та інструментів. Протягом усього циклу розробки було реалізовано як серверну, так і клієнтську частини системи, що забезпечують взаємодію користувача з функціоналом платформи. Створено інтерфейс користувача, який включає форми реєстрації та авторизації, а також можливість створення, перегляду та фільтрації заходів. Особливу увагу приділено зручності взаємодії з системою, структурованості даних, логіці обробки запитів, а також надійності зберігання інформації у базі даних PostgreSQL.

Розроблена система демонструє принципи організації публічних та приватних подій, онлайн-продажу квитків, надсилання сповіщень користувачам і можливість залишення відгуків про участь у заходах. Під час реалізації використовувалися сучасні технології, зокрема NestJS для бекенду, TypeORM для взаємодії з базою даних, а також Next.js для фронтенду, що забезпечило динамічну побудову сторінок і зручну навігацію [3].

Таким чином, виконання кваліфікаційної роботи дозволило не лише закріпити теоретичні знання, здобуті під час навчання, а й набути практичних навичок створення веб-систем, покращити розуміння архітектури клієнт-серверної взаємодії та сформуванати досвід повного циклу розробки сучасних веб-застосунків.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Гладунець М.І., Овчаренко Г.В. Розробка веб-застосунків: навчальний посібник. – Київ: Центр учбової літератури, 2021. – 256 с.
2. Grebenkin V. Modern Web Application Development: Best Practices and Insights. – Roller Books, 2024. <https://rollerbooks.com/rollerbooks-ukraine-web-development-books> (дата звернення: 13.05.2025)
3. Walke J. React – A JavaScript Library for Building User Interfaces. – Meta Platforms, 2025. <https://react.dev/> (дата звернення: 13.05.2025)
4. Dahl R. Node.js Documentation. – OpenJS Foundation, 2025. <https://nodejs.org/en/docs> (дата звернення: 13.05.2025)
5. Mozilla Contributors. Express/Node Introduction – Learn Web Development. – Mozilla Developer Network, 2025. https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs (дата звернення: 13.05.2025)
6. Otto M., Thornton J. Bootstrap Documentation. – Twitter, 2025. <https://getbootstrap.com/docs/5.3/getting-started/introduction/> (дата звернення: 13.05.2025)
7. Brown E. Web Development with Node and Express: Leveraging the JavaScript Stack. – O'Reilly Media, 2019. <https://www.amazon.com/Web-Development-Node-Express-Leveraging/dp/1491949309> (дата звернення: 13.05.2025)
8. Mammino L., Casciaro M. Node.js Design Patterns: Design and Implement Production-Grade Node.js Applications Using Proven Patterns and Techniques. – Packt Publishing, 2020. <https://www.amazon.com/Node-js-Design-Patterns-production-grade-applications/dp/1839214118> (дата звернення: 13.05.2025)

9. Hahn E. Express in Action: Writing, Building, and Testing Node.js Applications. – Manning Publications, 2016. <https://www.manning.com/books/express-in-action> (дата звернення: 13.05.2025)
10. Mezzalana L. Building Micro-Frontends: Architecting a Frontend Project for Scaling and Team Autonomy. – O'Reilly Media, 2021. <https://www.oreilly.com/library/view/building-micro-frontends/9781492082995/> (дата звернення: 13.05.2025)
11. Cherednichenko O., Kyrychenko I., Tereshchenko G., Miand D., Pylypenko S. Comparison of Blockchain-Based Data Storage Systems. – 2024. <https://doi.org/10.31110/COLINS/2024-3/010> (дата звернення: 31.05.2025)
12. Vysotska V., Kyrychenko I., Demchuk V., Gruzdo I. Holistic Adaptive Optimization Techniques for Distributed Data Streaming Systems. – 2024. <https://doi.org/10.31110/COLINS/2024-2/009> (дата звернення: 31.05.2025)
13. Kyrychenko I., Tereshchenko G., Smelyakov K. Optimized Indexing Method in a Hybrid Image Storage Model for Efficient Storage and Access in Big Data Environments. – 2024. <https://doi.org/10.1109/TCSET64720.2024.10755763> (дата звернення: 31.05.2025)
14. Halushka Maksym. GitHub profile. – GitHub, 2025. <https://github.com/NureHalushkaMaksym> (дата звернення: 13.05.2025)
15. Dashenkov D., Smelyakov K., Sharonova N. Dataset for NLP-enhanced image classification. – CEUR Workshop Proceedings, 2023, 3396, pp. 88–101. <http://ceur-ws.org/Vol-3396/> (дата звернення: 25.05.2025)
16. Smelyakov K., Hurova Y., Osiiivskyi S. Analysis of the Effectiveness of Using Machine Learning Algorithms to Make Hiring Decisions. – CEUR Workshop Proceedings, 2023, 3387, pp. 77–92. <http://ceur-ws.org/Vol-3387/> (дата звернення: 25.05.2025)

17. Byzkrovnyi O., Chupryna A., Smelyakov K., Sharonova N., Repikhov V. Comparison of Object Detection Algorithms for the Task of Detecting Possible Road Accident. – CEUR Workshop Proceedings, 2023, 3387, pp. 13–28. <http://ceur-ws.org/Vol-3387/> (дата звернення: 26.05.2025)

18. Shubin I., Kozyriev A. Method for Solving Quantifier Linear Equations for Formation of Optimal Queries to Databases. – CEUR Workshop Proceedings, 2023, 3396, pp. 449–459. <http://ceur-ws.org/Vol-3396/> (дата звернення: 21.05.2025)

Додаток А

Результат перевірки роботи на плагіат



Дата звіту 6/1/2025
Дата редагування ---

Звіт не був оцінений

Звіт подібності

метадані

Назва організації
Kharkiv National University of Radio Electronics
 Заголовок
2025_Б_ПІ_ПЗПІ-21-4_Галушка_М_В_скорочений
 Автор Науковий керівник / Експерт
Галушка Максим ВіталійовичСвєн Кардаш
 підрозділ
каф. ПІ

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.


Заміна букв		1
Інтервали		0
Мікропробіли		0
Білі знаки		2
Парафрази (SmartMarks)		11

Подібності за списком джерел


Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

Додаток Б

Слайди презентації



МІНІСТЕРСТВО
ОСВІТИ І НАУКИ
УКРАЇНИ




ХАРКІВСЬКИЙ
НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ
РАДІОЕЛЕКТРОНИКИ

Програмна система для управління заходами

Виконав:
ст. гр. ПЗП-21-4
Галушка М.В.

Керівник:
доцент кафедри ПП
Ворочек О.Г.



10 червня 2025

Мета роботи

Метою роботи є створення програмної системи для зручного управління заходами через веб-інтерфейс з підтримкою ролей користувачів. Система забезпечує швидкий доступ до інформації про події, автоматизує рутинні процеси: реєстрація, повідомлення та використовує сучасний стек технологій (React, Node.js, Express, PostgreSQL). Актуальність теми обумовлена зростанням потреби в онлайн-інструментах для організації подій, конференцій і семінарів.

Аналіз проблеми

Платформа	Переваги	Недоліки
Eventbrite	<ul style="list-style-type: none"> - Легке створення подій - Продаж квитків - Email-маркетинг 	<ul style="list-style-type: none"> - Обмежена кастомізація - Складність інтеграції з іншими системами
Ticketmaster	<ul style="list-style-type: none"> - Надійна система продажу квитків - Підтримка масових заходів - Розширені платіжні опції 	<ul style="list-style-type: none"> - Обмежене управління подіями - Не підходить для невеликих чи внутрішніх подій
Bizzabo	<ul style="list-style-type: none"> - Потужна аналітика - Підтримка корпоративних мережеских заходів - Багатофункціональність 	<ul style="list-style-type: none"> - Висока вартість - Складність використання для малого/середнього бізнесу

Постановка задачі та опис системи

Розробити веб-програму для ефективного управління подіями, яка дозволяє створювати, редагувати, видаляти та переглядати заходи з підтримкою різних ролей користувачів з відповідними правами доступу.

Система має забезпечувати централізоване зберігання даних про події та користувачів, надавати зручний та інтуїтивно зрозумілий інтерфейс, автоматизувати ключові процеси та сприяти ефективній організації як онлайн, так і офлайн-заходів.

Вибір технологій розробки

NestJS



Next.js

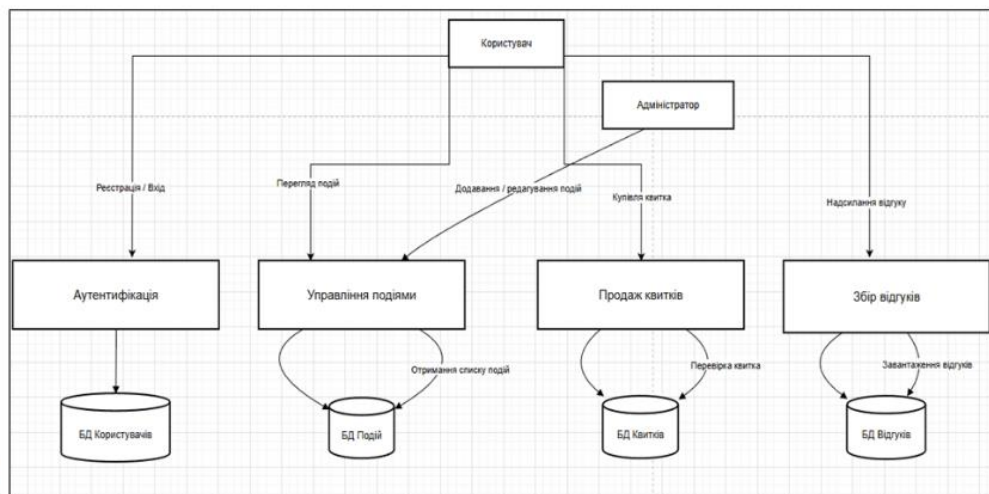


PostgreSQL



5

Архітектура створеного програмного забезпечення



6

Опис програмного забезпечення, що було використано у дослідженні

Розробка велася за клієнт-серверною архітектурою: спочатку створено структуру БД, потім реалізовано серверну логіку (авторизація, обробка запитів, керування подіями), і на завершення – адаптивний фронтенд. Використані технології: TypeScript (фронтенд/сервер), NestJS (REST API), TypeORM (PostgreSQL), Next.js (фронтенд), PostgreSQL (БД). Інструменти: GitHub (контроль версій), Visual Studio Code (IDE).



Дизайн системи

Методи:

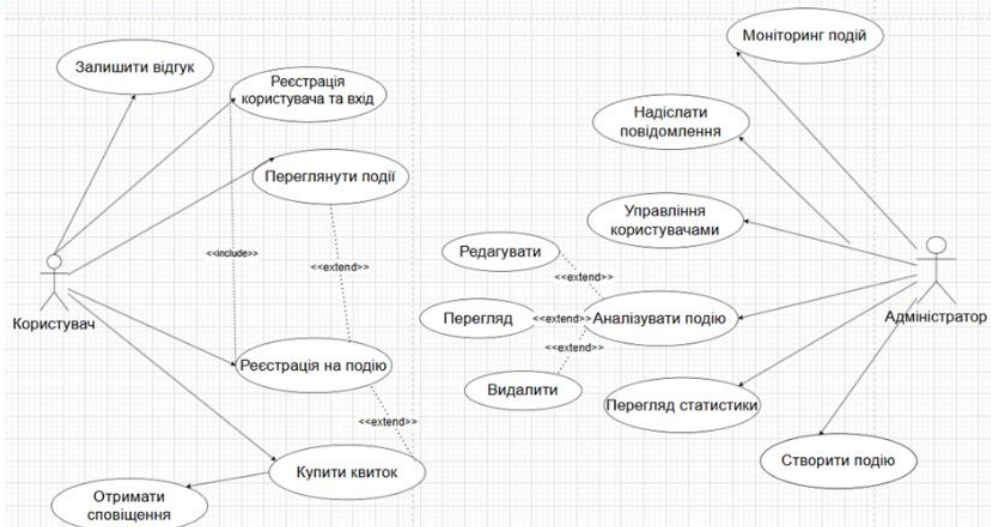
- компонентно-орієнтований підхід до створення інтерфейсу;
- REST-архітектура для взаємодії між фронтендом і бекендом;
- реляційна модель для зберігання даних;

Послідовність розробки:

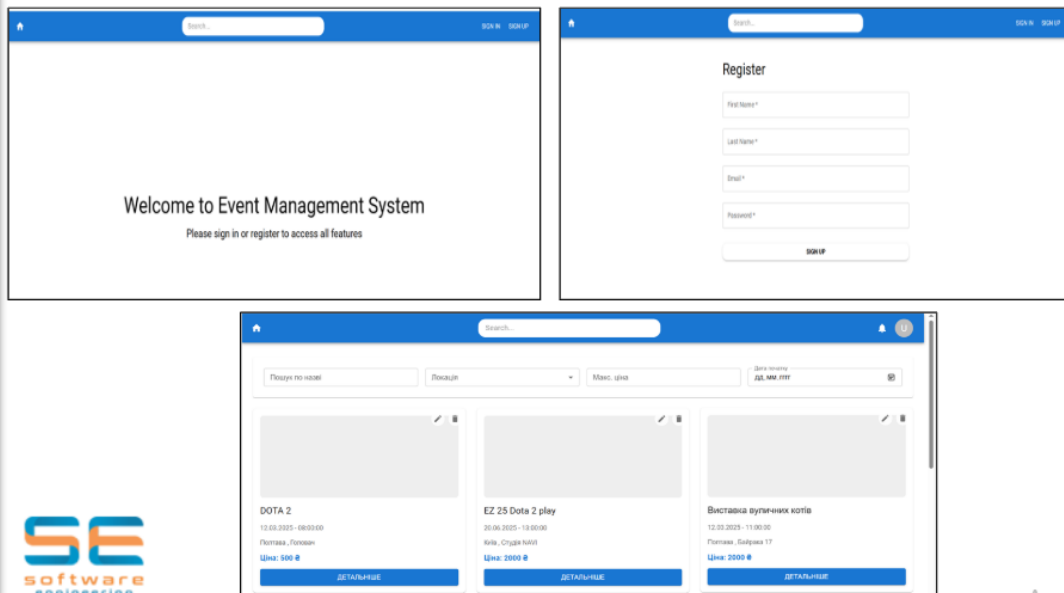
- проєктування бази даних та визначення ролей користувачів;
- реалізація API на NestJS (бекенд);
- розробка клієнтської частини на Next.js (фронтенд);
- інтеграція системи авторизації та сповіщень;
- тестування та налаштування адаптивного дизайну;



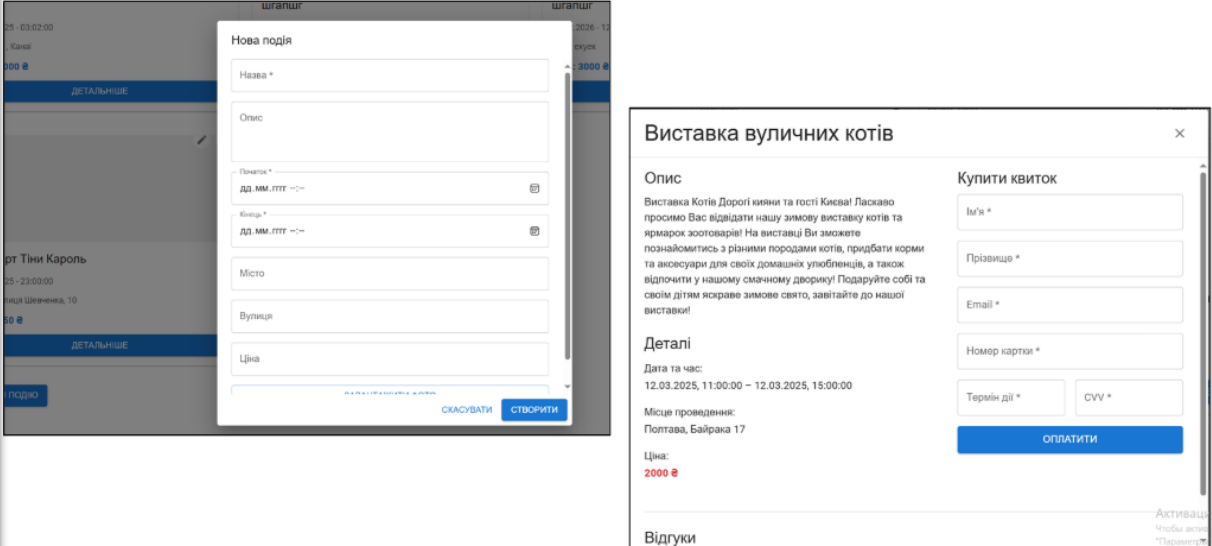
Приклад реалізації



Інтерфейс користувача



Інтерфейс користувача



The screenshot displays two main components of the user interface:

- Нова подія (New Event) Form:** A form for creating a new event with fields for:
 - Назва* (Name*)
 - Опис (Description)
 - Початок* (Start) with a date-time picker (ДД.ММ.ГГГГ - ЧЧ:ММ)
 - Кінець* (End) with a date-time picker (ДД.ММ.ГГГГ - ЧЧ:ММ)
 - Місто (City)
 - Вулиця (Street)
 - Ціна (Price)
- Виставка вуличних котів (Street Cat Exhibition) Details and Ticket Purchase:**
 - Опис (Description):** Виставка Котів Дорогі кишини та гості Києва! Ласкаво просимо Вас відвідати нашу зимову виставку котів та ярмарок зоотоварів! На виставці Ви зможете познайомитись з різними породами котів, придбати корми та аксесуари для своїх домашніх улюбленців, а також відпочити у нашому смачному дворіку! Подаруйте собі та своїм дітям яскраве зимове свято, завітайте до нашої виставки!
 - Деталі (Details):**
 - Дата та час: 12.03.2025, 11:00:00 – 12.03.2025, 15:00:00
 - Місце проведення: Полтава, Байрака 17
 - Ціна: 2000 ₴
 - Купити квиток (Buy Ticket):**
 - Ім'я* (Name*)
 - Прізвище* (Surname*)
 - Емейл* (Email*)
 - Номер картки* (Card Number*)
 - Термін дії* (Validity*)
 - CVV*
 - ОПЛАТИТИ (Pay)
 - Відгуки (Reviews):** Section for user reviews.

SE software logo is visible in the bottom left corner of the screenshot.

11

Тестування

Тест-кейс №1: Створення нового заходу авторизованим користувачем

Мета: Перевірити коректність функції створення заходу.

Передумови:

1. Відкрити веб-застосунок. (Пройдено)
2. Авторизуватися (логін/пароль) та бути перенаправленим на головну сторінку. (Пройдено)

Кроки тестування:

1. Натиснути «Створити захід» → відкривається форма. (Пройдено)
2. Ввести назву, опис, дату/час, вартість, місто та адресу. (Всі поля коректно заповнюються та перевіряються). (Пройдено)
3. Натиснути «Зберегти» → подія успішно створена, з'являється повідомлення. (Пройдено)
4. Перейти до перегляду події через «Докладніше» → сторінка події відображає введені дані. (Пройдено)

Результат тестування: ПРОЙДЕНО (P) **Тестувальник:** Галушка М.В. **Дата:** 28.05.2025



12

Тестування

Тест-кейс №2: Придбання квитка на захід

Мета: Перевірити успішне придбання квитка авторизованим користувачем.

Передумови:

1. Користувач має обліковий запис. (Пройдено)

Кроки тестування:

1. Увійти в систему. (Пройдено)
2. Перейти на головну сторінку заходів. (Пройдено)
3. Вибрати захід зі списку та натиснути «Детальніше». (Пройдено)
4. Відкривається форма покупки квитка. Заповнити поля: ім'я, прізвище, email, номер картки. (Пройдено)
5. Підтвердити оплату → з'являється повідомлення про успішну оплату. (Пройдено)
6. Придбаний квиток присутній у списку. (Пройдено)

Результат тестування: ПРОЙДЕНО (P) **Тестувальник:** Галушка М.В. **Дата:** 29.05.2025



Підсумки

Кваліфікаційна робота дозволила отримати практичний досвід розробки повнофункціонального клієнт-серверного веб-застосунку для управління подіями та продажу квитків. Було реалізовано зручний інтерфейс користувача та надійна серверна логіка з використанням сучасних технологій. Це закріпило теоретичні знання та покращило розуміння повного циклу розробки веб-систем.

